

MÓDULO PROFESIONAL DE DESARROLLO DE FUNCIONES EN EL SISTEMA INFORMÁTICO

Familia Profesional: Informática

Ciclo Superior de Administración de Sistemas Informaticos (CS ASI)

4 y 5 Trimestres

Profesor: Cesáreo García Rodicio

Fecha: Septiembre 2005

Índice de contenido

1 INTRODUCCIÓN.....	5
2 CONTEXTUALIZACIÓN.....	6
3 CAPACIDADES TERMINALES.....	8
4 REQUISITOS PREVIOS.....	11
5 CRITERIOS DE SECUENCIACIÓN DE LOS CONTENIDOS.....	12
6 ELEMENTOS CURRICULARES DE CADA UNIDAD DE TRABAJO.....	15
<i>Unidad Didáctica 1 Uso de Internet en el Desarrollo: documentación y búsqueda</i>	<i>15</i>
1.1 Contenidos Conceptuales.....	15
1.2 Contenidos Procedimentales.....	17
1.3 Criterios de Evaluación.....	18
<i>Unidad Didáctica 2 Lenguajes de Programación: Aprender un lenguaje nuevo.....</i>	<i>18</i>
2.1 Contenidos Conceptuales.....	18
2.2 Contenidos Procedimentales.....	21
2.3 Criterios de Evaluación.....	22
<i>Unidad Didáctica 3 Necesidades y Desarrollo de Nuevas funcionalidades.....</i>	<i>22</i>
3.1 Contenidos Conceptuales.....	22
3.2 Contenidos Procedimentales.....	25
3.3 Criterios de Evaluación.....	26
<i>Unidad Didáctica 4 Lenguajes Intepretados: Shell Bash.....</i>	<i>27</i>
4.1 Contenidos Conceptuales.....	27
4.2 Contenidos Procedimentales.....	29
<i>Unidad Didáctica 5 Publicación Web: Lenguaje HTML.....</i>	<i>30</i>

5.1 Contenidos Conceptuales.....	30
5.2 Contenidos Procedimentales.....	32
5.3 Criterios de Evaluación.....	33
<i>Unidad Didáctica 6 Publicación Web: Lenguaje CSS.....</i>	<i>33</i>
6.1 Contenidos Conceptuales.....	34
6.2 Contenidos Procedimentales.....	35
6.3 Criterios de Evaluación.....	35
<i>Unidad Didáctica 7 Arquitectura Web: Servidor Web Apache.....</i>	<i>36</i>
7.1 Contenidos Conceptuales.....	36
7.2 Contenidos Procedimentales.....	38
7.3 Criterios de Evaluación.....	40
<i>Unidad Didáctica 8 Desarrollo de Aplicaciones Web: páginas dinámicas.....</i>	<i>40</i>
8.1 Contenidos Conceptuales.....	41
8.2 Contenidos Procedimentales.....	42
8.3 Criterios de Evaluación.....	42
<i>Unidad Didáctica 9 Análisis, Diseño e Implantación de un proyecto Real (Aplicación Web).....</i>	<i>43</i>
9.1 Contenidos Conceptuales.....	43
9.2 Contenidos Procedimentales.....	44
9.3 Criterios de Evaluación.....	44
7 CONTENIDOS ACTITUDINALES.....	44
8 METODOLOGÍA.....	45
9 MEDIDAS DE ATENCIÓN A LA DIVERSIDAD.....	48
10 EVALUACIÓN.....	49
10.1 PROCEDIMIENTO DE EVALUACIÓN ORDINARIO	49

10.2 CRITERIOS Y PROCEDIMIENTOS PARA ESTABLECER LAS ACTIVIDADES DE RECUPERACIÓN DE LOS MÓDULOS PENDIENTES.....	52
10.3 PROCEDIMIENTOS DE EVALUACIÓN Y CUALIFICACIÓN EXTRAORDINARIA DEL ALUMNADO CON MÓDULOS PENDIENTES.....	53
10.4 CONTENIDOS MÍNIMOS EXIGIBLES.....	53
11 BIBLIOGRAFÍA.....	54

1 Introducción

Esta programación está referida al módulo profesional de Desarrollo de Funciones en el Sistema Informático (DFSI) que se cursa en el 2º año del ciclo formativo de grado superior de Administración de Sistemas Informáticos, concretamente en el cuarto y quinto trimestres. Este ciclo de grado superior tiene una duración total de 2000 horas (equivalente a 5 trimestres de formación en centro educativo como máximo, más la formación en centro de trabajo correspondiente).

La programación didáctica se desarrolla siguiendo las prescripciones del currículo publicado en el diario oficial de Galicia (DOGA) que reúne los siguientes aspectos:

- Contextualización
- Objetivos.
- Contenidos y secuenciación.
- Metodología.
- Criterios y procedimientos de evaluación.
- Atención a los alumnos con necesidades educativas específicas.
- Bibliografía de aula y de departamento.

Esta programación didáctica para un módulo forma parte de la programación general o proyecto curricular de ciclo, que los profesores del Departamento de Informática elaboran con el fin de dar unidad y coherencia a las enseñanzas de los módulos del ciclo formativo,

teniendo en cuenta las directrices establecidas en el Proyecto Curricular del Centro.

El Proyecto Curricular del ciclo formativo constituye el instrumento pedagógico-didáctico que articula a largo plazo el conjunto de actuaciones educativas del centro y tiene como objetivo alcanzar las finalidades educativas del mismo.

Según la propuesta en el DCB este módulo consta de 155 horas. Podemos tomar 5 horas de libre disposición (de entre las 90 que el centro puede escoger) quedando por lo tanto en 160 horas totales.

A partir del análisis del currículo, se han elaborado una serie de unidades de trabajo que han sido organizadas y secuenciadas en torno a supuestos prácticos, en base a los cuales se justifica el estudio de los contenidos básicos permitiendo relacionarlos con su aplicación práctica. Estos contenidos determinan las etapas más significativas de los procedimientos que se van a desarrollar (presentación, análisis y aplicación) en función de las variables diferenciadoras elegidas.

Por último, se han definido los **elementos curriculares** de cada unidad de trabajo, es decir, los objetivos que deben alcanzarse, los contenidos que deben desarrollarse, las actividades del proceso de enseñanza-aprendizaje que deben realizarse, la metodología a emplear y los criterios de evaluación que permitan medir el nivel de aprendizaje obtenido.

2 Contextualización

El diseño curricular del módulo pertenece al último nivel de concreción del Curriculum. Partiendo del Diseño Base del Título de Formación Profesional, que recordemos es

Técnico Superior en Administración de Sistemas Informáticos, se define el ciclo dentro del Plan Educativo del Centro y el Proyecto Curricular de Centro.

Vamos a contextualizar la programación en un centro educativo en zona urbana, bien comunicada y de un tamaño medio. Tiene los dos ciclos superiores de Informática, un ciclo superior de Administración y Finanzas y dos ciclos medios de Comercio. Además tiene programas de Garantía Social. La implantación de los ciclos superiores de informática son de reciente creación y sólo existe una promoción de técnicos superiores.

El centro pretende imprimir el carácter innovador, de educación “a medida” y adecuada al contexto laboral que se pretende imbuir en la “nueva formación profesional”. El centro pertenece a una ciudad urbana e industrial (Vigo) con presencia de empresas dedicadas al sector servicios informáticos de desarrollo de software y soporte técnico a infraestructuras informáticas. Las empresas demandan especialmente alumnos con formación en la arquitectura de Internet y administración de sistemas informáticos Windows y Linux.

En el tercer nivel de concreción curricular el ciclo formativo se coordinará con el otro ciclo de informática. El departamento de Informática ha decidido que el ciclo de DAI será por la mañana y el de ASI por la tarde, en grupos de máximo 30 alumnos y con al menos un ordenador por cada dos alumnos. Los locales asignados al ciclo cumplen en cuanto a instalaciones y dotación, los requisitos establecidos en el D.C.B., en el epígrafe 4.2. Requisitos mínimos de espacios e instalaciones para impartir estas enseñanzas.

Este módulo por lo tanto, se dará en horario de tarde lo que significa que los alumnos, en general, serán adultos y también que probablemente ya trabajen y algunos tengan

problemas para asistir a clase.

Dentro de la contextualización en el centro y el ciclo formativo se ha visto que el uso de Internet como plataforma integral en la administración de sistemas informáticos es necesario en el alumno. Por lo tanto, una particularización del curriculum será centrarse en la plataforma Internet-Web como soporte para el desarrollo de nuevas funcionalidades, soporte a los usuarios y documentación-colaboración del desarrollo.

Además el centro quiere buscar en el uso de las nuevas tecnologías de la información, ,especialmente con la colaboración a través de Internet, una formación personalizada y adaptada a cada alumno.

3 Capacidades Terminales

Este módulo está asociado a la unidad de competencia número 4: Proponer y Coordinar cambios para mejorar la explotación del sistema y las aplicaciones. Y tiene como capacidades terminales elementales:

- Aplicar el procedimiento de desarrollo y prueba de programas en lenguaje de tercera generación
- Manejar la documentación del programador e interpretar la información que suministra
- Manejar los sistemas operativos monousuario, multiusuario y de red y los sistemas gestores de bases de datos
- Aplicar técnicas de descripción de diseño de sistemas: diagrama de flujo de datos,

tablas y diagramas de transición de estados

- Detectar la composición actual del sistema: hardware, software, características y configuración, organización de datos almacenados
- Aplicar el interfaz de programación seleccionando las funciones adecuadas y definiendo correctamente las estructuras de datos necesarias.
- Analizar e interpretar las especificaciones de problemas de explotación, formulados sobre un sistema.
- Documentar la configuración del sistema y los requisitos de cambio presentados..
- Redactar informes sobre distintas propuestas de cambios en el sistema.
- Diseñar algoritmos que respondan a las propuestas especificadas y que utilicen adecuadamente las técnicas de programación sobre el sistema: programación concurrente, sincronización y comunicación de procesos.
- Seleccionar el entorno de programación e implementación del diseño más adecuado a la naturaleza del problema presentado
- Elaborar y diseñar juegos de prueba acordes a los requerimientos que se van a probar y aplicar metodologías de diseño sistemático de datos de prueba.
- Formular especificaciones de requerimientos sobre los programas y elementos del sistema, necesarios para responder a los problemas propuestos.
- Proponer distintas soluciones de cambios a realizar sobre el sistema y su configuración según los requerimientos propuestos.

- Evaluar si el juego de datos desarrollado está de acuerdo con las especificaciones propuestas
- Evaluar las incidencias de las soluciones proyectadas y los programas desarrollados sobre la funcionalidad y el rendimiento del sistema y su adecuación a las especificaciones propuestas.
- Evaluar si la documentación producida y adecuada, según sus objetivos, en la forma y en el contenido

Resumiendo y sintetizando todas las capacidades terminales que iremos evaluando, tendríamos como objetivos del módulo:

- Aprender técnicas de programación, especialmente con lenguajes interpretados, que permitan mejorar los procedimientos de explotación del sistema
- Manejar con soltura la Arquitectura Web para el desarrollo de nuevas funcionalidades sobre el SO, para la documentación utilizando publicación web y como soporte técnico a los usuarios.
- Generar documentación en el sistema sencilla, disponible en Internet y fácilmente actualizable.
- Entender el proyecto de desarrollo como un todo, desde los requisitos del cliente-usuario hasta la puesta en marcha.

El contenido de este módulo es sobre todo de tipo procedimental y el alumno deberá adquirir una serie de destrezas y habilidades en el manejo y administración del sistema. Pretendemos que el módulo sea una propuesta interesante y actual para el alumno.

Los contenidos básicos de tipo actitudinal irán orientados a fomentar la autonomía, el aprendizaje de nuevos recursos (lenguajes) y la necesidad de una formación continua como síntesis de los descritos en el Diseño Curricular Base.

4 Requisitos Previos

El alumno tendrá la capacitación prevista en los módulos de los primeros tres trimestres del ciclo. Estas competencias están descritas en los módulos correspondientes y resumiéndolas podríamos definir los siguientes:

- Administración y Gestión de la conectividad de Red, especialmente una TCP/IP
- Administración y Gestión Básica/Media de un sistema operativo multiusuario y multitarea
- Conocimientos de programación en un lenguaje estructurado, especialmente lenguaje C y herramientas de análisis y diseño (básicas).
- Conocimiento medio de un editor de texto avanzado

Además son de interés todos los contenidos conceptuales, procedimentales y actitudinales de todas las asignaturas de primer curso. Se prestará atención a los alumnos que tienen módulos pendientes o con notas iguales o inferiores al 6.

Se hará una evaluación inicial del perfil de la clase, para definir correctamente el contexto del grupo. Al ser el segundo curso del ciclo es más fácil que el grupo sea más homogéneo en conocimientos, sin embargo no debemos olvidar que en áreas como la informática existe un alto nivel de alumnos con formación autodidacta y que incluso en cursos de

verano han podido alcanzar algunos objetivos del módulo.

5 CRITERIOS DE SECUENCIACIÓN DE LOS CONTENIDOS

Los criterios generales de secuenciación adoptados son:

- Identificar los principales componentes de los contenidos seleccionados en el primer nivel de concreción (DCB –Diseño Curricular Base-)
- Analizar las relaciones que existen entre los componentes identificados y establecer las estructuras de contenido correspondientes
- Proponer una secuenciación de los componentes atendiendo a las relaciones y estructuras establecidas y a las leyes del aprendizaje significativo

Los contenidos se organizan en varios bloques distribuidos secuencialmente. Las unidades didácticas se pueden englobar en cuatro grandes bloques y una unidad didáctica (UD) de transición:

- Aprender un nuevo lenguaje de programación (UDs 1,2,3 y 4)
- Arquitectura Web para el desarrollo y la documentación de nuevas funcionalidades (UDs 5,6,7 y 8)
- Desarrollo de un mini proyecto (UD 9)

Esta distribución en bloques permite una comprensión general del módulo, pero nos centraremos en una programación por unidad didáctica pues permite un mejor control y

evaluación.

Aunque hemos hecho la distribución por horas, en la práctica una clase es un conjunto de sesiones de 50 minutos, generalmente dos o tres sesiones. En realidad es más útil el concepto de clase para el docente, pues para el alumno englobaría el concepto de “lo que voy a ver en la clase de hoy”. Obviamente sólo la experiencia permite determinar exactamente cuales son los contenidos para cada clase, sin embargo es imprescindible tener una programación secuencial según el número de horas teniendo en cuenta las clases a impartir.

En realidad un trimestre consta aproximadamente de unas 12 semanas. Por lo tanto el módulo tiene una duración aproximada de 24 a 28 semanas. Según el diseño curricular base del ciclo más 5 horas de libre disposición del centro, el módulo consta de 160 horas en total. Esto significa 9 sesiones de 50 minutos, y según considere el centro se necesitarán más o menos semanas para dar el módulo. Las unidades didácticas serán impartidas siguiendo la secuencia siguiente:

UD	TÍTULO	Semanas	Clases	50 min
1	Uso de Internet en el Desarrollo: documentación y búsqueda	2	4	8
2	Lenguajes de Programación: Aprender un lenguaje nuevo	2	4	8
3	Necesidades y Desarrollo de Nuevas funcionalidades	2	4	8
4	Lenguajes Intepretados: Shell Bash	4	8	28

UD	TÍTULO	Semanas	Clases	50 min
5	Publicación Web: Lenguaje HTML	2	4	14
6	Publicación Web: Lenguaje CSS	3	6	21
7	Arquitectura Web: Servidor Web Apache	3	6	21
8	Desarrollo de Aplicaciones Web: páginas dinámicas	2	4	14
9	Análisis, Diseño e Implantación de un proyecto Real (Aplicación Web)	5	25	70
		160	25	65
				192

El módulo está dividido en dos evaluaciones en los dos primeros trimestres del segundo curso del Ciclo Superior, es decir el cuarto y quinto trimestre del ciclo. La secuenciación de las unidades didácticas debería permitir comenzar la segunda evaluación con el trabajo de desarrollo propiamente dicho, esto significaría que en el cuarto cuatrimestre se comenzaría con la UD 9.

Esta distribución es meramente orientativa, dado el carácter abierto y flexible del currículo, en función de las necesidades del grupo podrán producirse pequeños ajustes.

6 Elementos Curriculares de cada Unidad de Trabajo

Unidad Didáctica 1 Uso de Internet en el Desarrollo: documentación y búsqueda

En esta unidad didáctica es la primera del módulo, esto significa que los alumnos vienen de su descanso vacacional y que se debe empezar con contenidos más prácticos para introducirlos en el ritmo del trimestre. Esta unidad continuará con el trabajo de los módulos anteriores para el uso de Internet en clase, haciendo hincapié en el uso de Internet para el desarrollo de software, la documentación y la ayuda en línea.

En todo el módulo el uso de Internet estará presente como ayuda y documentación, como publicación y colaboración con el equipo de trabajo y finalmente como arquitectura de desarrollo.

En esta unidad de trabajo se utilizan exposiciones en clase, grupos de Yahoo, soporte de colaboración vía web de la clase y los servicios instalados en el centro (Cliente POP y Cliente HTTP)

1.1 Contenidos Conceptuales

1 Internet

1.1 Introducción

1.1.1 Nociones: Sockets, Puertos, Repaso de Redes TCP/IP

1.1.2 Colaboración por Internet para el módulo

1.1.3 Plataforma de Apoyo a las clases vía Web o grupos Yahoo

1.2 Descripción de recursos en Internet (URL Avanzado)

2 Correo Electrónico Avanzado

2.1 Opciones de Configuración: servidores, redireccionamiento, autenticación, etc

2.2 Configurar y Gestionar Carpetas

2.3 Importar y Exportar Mensajes de Correo Electrónico

2.4 Configuración de Varias Cuentas de Correo

2.5 Consulta de múltiples cuentas desde diferentes lugares

3 Buscar Información

3.1 Uso Básico de Google

3.2 Búsqueda Avanzada en Google

3.3 Formatos de Archivo: txt, html, pdf, sxw y doc

3.4 Búsqueda en listas de noticias

4 NetEtiquette

4.1 Normas de Educación en el envío de Mensajes

4.2 Relación Virtual: disponer de una identidad en Internet

4.3 Generación de confianza: escoger correctamente el nombre de usuario

5 Plataforma de Colaboración (grupos Yahoo):

5.1 Qué son

5.2 Cómo y para qué lo utilizaremos en clase

5.3 Servicios que nos ofrece

1.2 Contenidos Procedimentales

1. Gestión de una cuenta de Correo Electrónico

a) Crear una cuenta de correo electrónico gratuita. Conexión WebMail y POP

b) Configuración en un ordenador del centro

c) Manual de configuración de la cuenta de correo electrónico del colegio

2. Buscar información:

a) Buscar información sobre un lenguaje de programación

b) Crear un archivo de texto con links (bookmarks) interesantes y su descripción

c) Añadir enlaces de Internet a Bookmarks o Mis Favoritos

d) Crear y Gestionar una cuenta de noticias (ejemplo lenguaje)

3. Gestión del Grupo Yahoo y/o Comunidad Colaboración

a) Solicitar el ingreso en el grupo Yahoo y participar en él (mensaje de bienvenida)

b) Enviar mensaje desde el portal Web y desde el correo electrónico

1.3 Criterios de Evaluación

- Ser capaz de manejar el correo electrónico de una manera eficiente
- Ser capaz de realizar búsquedas avanzadas con un buscador (p.ej. Google)
- Ser capaz de colaborar en la plataforma de colaboración escogida

Unidad Didáctica 2 Lenguajes de Programación: Aprender un lenguaje nuevo

En esta unidad se pretende que el alumno/a entienda el esquema de trabajo ante un lenguaje de programación nuevo. Se pretende que el alumno/a comprenda la necesidad de tener en cuenta aspectos diferentes para el desarrollo de un programa, desde la plataforma de trabajo hasta el software necesario para utilizar realizar el primer programa.

Los contenidos son básicamente conceptuales y procedimentales aunque están inmersos los contenidos actitudinales en el trabajo en equipo. Las habilidades que intervienen son de interpretación, análisis y síntesis de la información necesaria para buscar y comprender cada uno de los elementos necesarios para crear un programa independiente del lenguaje.

2.1 Contenidos Conceptuales

1 Aspectos a tener en cuenta para comenzar a programar

1.1 Plataforma de Trabajo: Arquitectura Hardware y Sistema Operativo

1.2 Documentación en diferentes soportes: impreso y en Internet

1.3 Lenguaje y Recursos Extra (librerías)

1.4 Entorno de Desarrollo: Herramienta de Trabajo

1.5 Criterios para escoger un lenguaje de programación

2 Arquitectura del Computador (Repaso)

2.1 Máquina de Von Neumann

2.2 Arquitecturas y Plataformas Hardware de desarrollo

2.3 Programas Interpretados y Compilados

3 Qué es un programa

3.1 Programas Fuente: archivos de texto

3.2 Control y numeración de versiones

3.3 Documentación

4 Características Generales de un Lenguaje

4.1 Caracteres Básicos y Palabras Reservadas

4.2 Operadores Aritméticos y Lógicos

4.3 Sentencias de Control

4.4 Sentencias de Repetición

4.5 Programación Modular y Estructurada

4.5.1 Funciones, Módulos y Librerías

4.5.2 Documentación y Conexión

4.6 Programación Orientada a Objetos

4.6.1 Descripción básica de un objeto: coche

4.6.2 Lenguajes Orientados a Eventos como un ejemplo simplificado de POO

5 El Lenguaje de Programación

5.1 Cómo utilizar los datos: Alfabetos y Códigos. Sistemas de Codificación de texto

5.2 Niveles: Léxico (símbolos), Sintáctico (construcciones) y Semántico (significado)

5.3 Elementos del Lenguaje: Identificadores, Constantes, Operadores, Instrucciones y Comentarios

5.4 Características: Universalidad, Detallabilidad, Generalidad, Usabilidad

6 Clasificación de los Lenguajes

6.1 Cercanía a la Máquina: máquina, ensamblador, alto nivel

6.2 Evolución del Hardware: 1a (40-60), 2a (60-64), 3a (64-72), 4a (72-81), 5 (81-90), 6a (Internet)

6.3 Estructura Gramatical: Imperativo (digo cómo hacerlo) y Declarativo (digo qué hacer)

6.4 Según el procesado: interpretados y compilados

7 Uso del lenguaje

7.1 Procesado: análisis léxico, análisis sintáctico, análisis semántico, generar código y ejecutar código

7.2 Documentación, Manuales y Tutoriales: buscar los adecuados al lenguaje

7.3 Programación Estructurada y Modular

8 Ejemplo con un lenguaje: Descripción y Preparación del entorno

8.1 Ver la Plataforma de Desarrollo (Hardware y Software)

8.2 Buscar la Documentación y Soporte para el Aprendizaje y Programación

8.3 Buscar Entorno de Desarrollo (para la edición y desarrollo del programa)

8.4 Palabras Reservadas, Tipos de Datos, Asignación de Variables, Operadores Aritméticos y Lógicos, Herramientas de Control de Flujo, Iteraciones y Recursividad, Módulos (Procedimientos, Funciones), Parámetros y Estructura General

2.2 Contenidos Procedimentales

4. Escoger un lenguaje de programación para desarrollar en el módulo

5. Buscar algunas Alternativas de lenguajes (tres)

6. Analizar Ventajas e Inconvenientes (4) y contrastar

7. Tomar decisión por un lenguaje

8. Hacer una ficha con los requerimientos del lenguaje

9. Arquitectura Hardware: en clase y en casa

10. Sistema Operativo, versión del lenguaje y librerías necesarias

11. Necesidades para programar: en clase y en casa

12. Entorno de Desarrollo

13. Configurar entorno de desarrollo: programas que se necesita

14. Tener Instalación Local de Documentación y Links (Bookmarks)

15. Comenzar a trabajar con el lenguaje, sin objetivo concreto: historia, palabras reservadas, sintaxis más común, etc

16. Escribir un programa en el lenguaje y ver errores del intérprete

2.3 Criterios de Evaluación

- Tener criterio para escoger un lenguaje de programación
- Entender que partes son las necesarias para tener un entorno de desarrollo
- Ser capaz de configurar un entorno de desarrollo
- Entender los diferentes aspectos de un lenguaje de programación

Unidad Didáctica 3 Necesidades y Desarrollo de Nuevas funcionalidades.

¿Por qué necesitamos desarrollar nuevas funciones? ¿Son realmente necesarias nuevas funcionalidades? En esta unidad trataremos de responder a estas preguntas viendo las necesidades del análisis y el diseño, y aprovecharemos para repasar los comandos de consola Unix como un ejemplo de lenguaje básico. Se verá también la utilidad del uso de librerías y definición de interfaces entre programas. El modelo de desarrollo de tres capas (3-tier) es un modelo interesante para analizar el desarrollo de una nueva funcionalidad.

3.1 Contenidos Conceptuales

1 Sistema Informático

1.1 Hardware: plataformas típicas

1.2 Software: Sistema Operativo y Aplicaciones

1.3 Personas: usuarios finales y administradores

2 Modelo de Tres Capas

2.1 Presentación (Diseño)

2.2 Programación (Lógica del Negocio)

2.3 Datos (Modelo de Datos)

3 Necesidades para la Administración del Servidor

3.1 Automatizar tareas para los usuarios

3.2 Automatizar tareas de arranque y parada del sistema

3.3 Monitoreo del Sistema, Estadísticas e Informes

4 Programación de Propósito General

4.1 Requisitos del Cliente

4.2 Análisis de qué quiere

4.3 Diseño del programa o alternativas

4.4 Programación e Implementación

4.5 Puesta en Marcha y Documentación

5 Herramientas de Análisis y Diseño (repass)

5.1 Diagramas de Flujo

5.2 Diagramas de Procesos/Eventos

6 Configurar un Entorno de Desarrollo

6.1 Arquitectura y documentación

6.2 Tipos de Herramientas: básicas y avanzadas (CASE y Entornos IDE)

7 Características de un Editor de Textos Avanzado

7.1 Multiplataforma

7.2 Soporte a mi lenguaje de programación

7.2.1 Etiquetado/Colores

7.2.2 Tabulación e Indentación

7.3 Personalizable y Configurable

7.4 Buena Documentación

7.5 Gratuito y con comunidad de soporte

7.6 Soporte para automatización de tareas (macros)

7.7 Modular: programas para aumentar la funcionalidad

8 Funcionamiento (Sugerencia con Emacs aunque es válido cualquiera)

8.1 Gestión de Texto: añadir, quitar, modificar

8.2 Gestión de Archivos de Texto (programas)

8.3 Gestión de búsquedas y reemplazos

8.4 Gestión del lenguaje de programación: resaltado de palabras reservadas y sintaxis, control sintáctico (paréntesis, indentación) e integración con la documentación del lenguaje.

8.5 Soporte avanzado: varios archivos a la vez, soporte a control de versiones, conexión a consolas remotas, etc

9 Nociones básicas para trabajo con programas: librerías, diferentes lenguajes de programación, control de versiones y trabajo con otros programadores

3.2 Contenidos Procedimentales

1. Analizar un programa según modelo de tres capas

- a) Programa C del año pasado
 - b) Cliente de Correo Electrónico
 - c) Programa Kazaa
 - d) Página Web (Terra)
-

2. Ejemplo programa: gestionar la biblioteca del colegio

- a) Obtener los requisitos del proyecto
 - b) Criterios para obtener las necesidades del cliente
 - c) Documentación de la reunión
-

3. Diseño del programa

- a) Utilizar bloques de dibujo, para separar el programa en bloques
-

b) Utilizar diagramas de control de flujo

c) Planificar el Trabajo: usuarios, tiempo de desarrollo y división de tareas

4.Herramientas de Diseño

a) Flujo de Datos

b) Diagramas de Procesos y Eventos

5.Gestión de un Archivo de texto

a) Instalación y Puesta en Marcha

b) Gestión de varios archivos de texto en una sola consola

c) Insertar texto (copiar/cortar/pegar)

d) Realizar búsquedas y reemplazos

6.Generación de un informe texto con el editor (pasar un archivo de word a texto)

7.Gestión de programas con editor de texto

a) Revisar si tiene soporte para el lenguaje escogido

b) Revisar la personalización para el lenguaje: resaltado y control de sintaxis

c) Revisar si puede trabajar con el lenguaje C

8.Gestión de programa complejo en C (del primer año): librerías y programas fuente

3.3 Criterios de Evaluación

→ Ser capaz de Analizar un problema según el modelo de tres capas

- Plasmar en un documento las necesidades de un cliente
- Realizar un diseño de problemas sencillos utilizando diagramas
- Manejo de un editor de texto para funcionalidades básicas
- Manejo con el editor de un programa ejemplo
- Distinguir el soporte especializado del editor de texto para el uso con el lenguaje

Unidad Didáctica 4 Lenguajes Intepretados: Shell Bash

El uso de lenguajes interpretados frente a lenguajes compilados permite un mejor mantenimiento y un desarrollo más rápido. El uso de lenguajes de script es hoy en día muy común en el desarrollo de software.

El lenguaje del intérprete Bash es una herramienta muy potente para el desarrollo de utilidades de administración. Es la herramienta de trabajo principal de un administrador en consola Unix.

4.1 Contenidos Conceptuales

1 Introducción

1.1 Qué es un script

1.2 Primer Script Hello World! (archivo de texto con encabezado + ejecutable + exit)

1.3 Archivos de configuración de usuario: .bashrc, .profile, etc

1.4 Importancia del intérprete del shell (/bin/bash, o /usr/bin/perl, o /usr/bin/tcl)

2 Variables

2.1 Variables: asignación y tipos (VAR="hola" o VAR=3.14) y ámbito (scope) de las

variables)

2.2 Sustitución de variables $\${VAR}$

2.3 Variables Especiales: $\$#$, $\${0-9}$, $\$@$, $\$?$, $\$\$$, $\$!$

2.4 Variables de Entorno: LOGNAME, HOME, IFS, PATH, PS1, PS2, TERM, etc

2.5 Gestión de Variables de entorno: env, set, unset y export

3 Entrada y Salida de Datos

3.1 Redirección: stdin, stdout, stderr

3.2 Tuberías o Pipes: |

3.3 Entrada de Datos: comando read y línea de comandos ($\$0$, $\$1$...)

4 Sentencias de Control

4.1 Repetitivas : for...do...done, while...do...done

4.2 Flujo if ... then ...else ...fi, case value in ...esac

5 Comparación y Expresiones:

5.1 Sintaxis Test expression ó [expression] (para un operando o dos)

5.2 Ejemplos de Opciones: evaluación de cadenas (-z, -n, cadena1 != cadena2),
evaluación de archivos(-d, -x, -e, fichero1 -ot fichero2), evaluación aritmética (-lt, -eq
, -ge) y evaluación lógica (-o, -a)

6 Funciones

6.1 Qué son

6.2 Sintaxis: valores de entrada ($\$1$... $\$9$) y de salida (return)

6.3 Archivos de Funciones: Librerías

6.4 Cargar una librería: `./libreria.sh`

7 Varios

7.1 Caracteres de escape y Especiales (\n,\r,\t,\v,\b,\a,\\") de echo

7.2 Funcionamiento BackStick `: ejecutar comandos

7.3 Más sentencias: select (control de flujo), read (entrada)

7.4 Reglas de Expansión de llaves, de variables/parámetros (',"), de comandos/orden, aritmética \$((expresion)), división de palabras (IFS), caracteres comodín (patrones:[],*,?)

7.5 Comando shift: desplazar una lista de valores

7.6 Avanzado: experimentar con comando dialog

4.2 Contenidos Procedimentales

1 Uso básico de un script

1.1 Uso del manual en línea (páginas man e info)

1.2 Realizar un script (ejercicio1.sh) que muestre en pantalla la cadena "Hola" y otro que diga "Hola soy el usuario: <personalizado>".

2 Uso de sentencias de control

2.1 Realizar un script (ejercicio2.sh) que tome un número desde la consola y vaya escribiendo números de forma decreciente.

2.2 Realizar un script (ejercicio3.sh) que tome dos números (o cadenas) desde la consola y realice cuatro tipos de comparación con test

3 Uso de librerías

3.1 Realizar las funcionalidades de los scripts anteriores con funciones y cargarla desde el script (funciones.lib)

3.2 Intercambiar esta librería entre alumnos y grupos y tendrá que funcionar.

4 Utilidad para administrar el sistema

4.1 Realizar una librería de funciones (monitorear_sistema.lib) para revisar el uso de CPU, el uso de Memoria y el uso de los sistemas de ficheros.

4.2 Controlar los errores debido a falta de privilegios y dar los mensajes correspondientes al usuario

5 Realizar un programa que simule una agenda telefónica (dos últimas sesiones)

Unidad Didáctica 5 Publicación Web: Lenguaje HTML

En el planteamiento general del módulo utilizamos Internet como la plataforma de trabajo y colaboración. La documentación tanto del proceso de desarrollo como de la experiencia final de usuario se ha de apoyar en Internet como una plataforma ágil, sencilla y barata de utilizar.

Además, el lenguaje HTML, con el que están hechas las páginas web aunque no es un lenguaje de programación como tal está enfocado a lenguajes de descripción y estructuración de datos y permite al alumno ver la utilidad de este tipo de lenguajes. Esta unidad didáctica ofrecerá al alumno no sólo el conocimiento básico del lenguaje HTML sino también aspectos básicos de la publicación Web teniendo en cuenta sus características

5.1 Contenidos Conceptuales

1 Introducción a HTML. SGML (iso 8879) y XML

1.1 Diferencia entre Estructura y Contenido

1.2 Diferencia entre Contenido y Diseño

1.3 Documentar en Web: simplicidad y claridad

1.4 Características de un marcado HTML (etiqueta y atributos)

1.5 Entorno de Desarrollo: Editor de Textos

1.6 Recursos: estándares W3C e Internet

2 Estructura de un documento HTML

2.1 Documento: html

2.2 Encabezado: head (Título: title)

2.3 Cuerpo: body

3 Marcando el texto

3.1 Párrafos: p

3.2 Salto de línea: br

3.3 Resaltando texto (Negritas: b y em)

3.4 Encabezamientos: h1 hasta h6

3.5 Anclas, Nombres y Links: a

3.6 Link internos (con #) y externos

4 Construyendo Listas y Tablas

4.1 Tipos de listas (numeradas y no numeradas): ol, ul

4.2 Items de las Listas: li

4.3 Definición de Listas: dl, dt, dd

4.4 Tablas, filas, datos y encabezados: table, tr, th, td

5 Aprender: Ver el código fuente de una página Web

6 Diseñar las páginas web

6.1 Para qué la necesito

6.2 Secciones y Directorios

6.3 Mantenimiento

7 Publicación Web

7.1 Diferencias entre medio impreso y pantalla (on-line)

7.2 Criterios para publicar una página web

7.3 Criterios para el uso de imágenes y recursos audiovisuales

7.4 Concepto de navegabilidad vertical y horizontal

7.5 Relación con Internet: uso adecuado de los links

5.2 Contenidos Procedimentales

1. Primera Página Web

- a) Escribir una página de texto (index.txt) con tres o cuatro líneas de texto
- b) Escribir una página con estructura HTML (index.html) con la estructura básica y el texto anterior.

2. Formato de Texto

- a) Añadir algunos encabezados, varios párrafos con salto de línea
- b) Añadir links externos (a una página en un directorio y a una página externa)

3.Formato de tablas y listas

- a) Escribir dos tablas con varias columnas y filas
- b) Escribir ejemplo de los tres tipos de listas e intercambiar formato

4.Página Personal

- a) Proceso de Actualización y Mantenimiento (Colegio)
- b) Diseño en papel: objetivos, secciones y flujo de páginas

5.Página Personal (dos clases)

5.3 Criterios de Evaluación

- Ser capaz de diseñar una página web sencilla
- Ser capaz de mantener una página web personal

Unidad Didáctica 6 Publicación Web: Lenguaje CSS

Dentro del modelo de tres capas, utilizaremos el Lenguaje CSS para definir los estilos y formatos de presentación de la publicación Web. Es fácil de aprender y gracias a su fácil mantenimiento puedo modificar la presentación de un servicio Web completo. Esta unidad también permite ver la conexión entre dos lenguajes, HTML y CSS, ofreciendo al usuario final una sólo experiencia.

Tanto esta unidad como la anterior tienen pocos contenidos conceptuales y están orientados a contenidos de tipo procedimental que se verán reflejados en la realización de la página web personal del alumno.

6.1 Contenidos Conceptuales

1 Criterios para separar formato (CSS) de contenido (HTML)

1.1 Estándares W3C: CSS 2.0

1.2 Capa Presentación y Programación y Datos

2 Hojas de Estilos (CSS)

2.1 Qué son y para qué sirven

2.2 Aplicar estilos en una página web Etiqueta Link<link href="URL" rel="stylesheet" type="text/css">

2.3 Archivos CSS: archivos de texto

2.4 Características generales de las reglas: entender la sintaxis

2.5 Propiedades CSS 'nombre_de_la_propiedad': Valor, Inicial, Se aplica a, Se hereda, Porcentajes, Medios

2.6 Valor: palabras clave, datos básicos (medidas, fuentes), tipos de datos de una propiedad

2.7 La barra |, la doble barra ||, operadores * + {n,m} ? (opcional)

2.8 Ejemplos de propiedades del lenguaje (Ejemplo de Hoja de Estilos)

3 Sintaxis: Etiqueta { propiedad: valor [;...]} Análisis de varios ejemplos (3)

4 Ejemplo Básico de una hoja de estilos (default.css)

4.1 Clases para diferenciar estilos

4.2 Tipo de letra, tamaño y color

4.3 Colores y estilos de tablas

5 WebLog

5.1 ¿Qué son?

5.2 Utilidad (publicación sencilla) y Características Principales (diseño sencillo, publicación sin saber HTML y comentarios)

5.3 Sistemas de Publicación: Blogger (Básico) y Movable Type (Avanzado)

5.4 ¿Cómo lo pongo en marcha?. Funcionamiento: Administración/Usuario

6.2 Contenidos Procedimentales

1.Desarrollo de Página Personal

- a) Enlace de una página a una hoja de estilos
- b) Criterios sobre aspectos gráficos de mi página Web

2.Crear una hoja de estilos

- a) Analizar ejemplos de sintaxis (tres ejemplos, desde sencillo a más complicado)
- b) Diseñar un ejemplo de informe de incidencia
- c) Crear una hoja de estilos (informe.css). Utilizar clases

3.Puesta en Marcha de un WebLog

4.Trabajo sobre la página web personal (básico-medio) y publicación web de un archivo

Word Ejemplo

6.3 Criterios de Evaluación

→ Creación de diferentes hojas de estilos

→ Mantenimiento de la parte de diseño de mi página web personal

Unidad Didáctica 7 Arquitectura Web: Servidor Web Apache

El uso de un navegador (cliente HTTP) es bastante común por parte de cualquier alumno que haya utilizado Internet, sin embargo para la administración de un servicio Web hay que conocer la administración de un Servidor HTTP. En esta unidad didáctica nos fijamos en la administración del Servidor Apache, el más utilizado en Internet.

El contenido de esta unidad didáctica es predominantemente procedimental aunque es importante conocer algunos contenidos conceptuales centrados en el conocimiento básico del protocolo HTTP.

7.1 Contenidos Conceptuales

1 Protocolo HTTP

1.1 Estándar del protocolo

1.2 Nociones Básicas

1.3 Métodos más comunes GET y POST

1.4 Tipos de Errores

2 Servidores HTTP

2.1 ¿Qué es y qué hace un Servidor HTTP?

2.1.1 Procedimiento de envío de una página HTML

2.1.2 Problemas en la disponibilidad de páginas web

3 Servidor Web Apache

3.1 Historia del Proyecto y Lugar de Documentación

3.2 Estadísticas y Comentarios

4 Instalación de Apache

4.1 Plan de Trabajo: qué necesito, software necesario (versión), módulos y documentación

4.2 Cómo trabajar con los códigos fuente (tar, gz, bzip). Leer: README, INSTALL, etc

4.3 Instalación: configuración (./configure), compilación (make), instalación (make install)

5 Configuración y Puesta en Marcha

5.1 Archivo de Configuración: httpd.conf

5.2 Arrancar Apache (apachectl start, httpd -f httpd.conf, kill `cat apache.pid`, inittab/rc.d)

5.3 Probar que está funcionando: acceso (telnet y/o navegador), ver procesos (padre e hijos) y parar/reiniciar

5.4 Revisión de Logs: interpretación de líneas y tipos de formatos

5.5 Problemas: revisar logs (access.log, error.log) Comando tail

6 Configuración Básica

6.1 Archivo de Configuración: httpd.conf (Documentación on-line y en Archivo)

6.2 Forma de Configuración (Directivas). Sintaxis: Syntax, Default, Context, Override, Status, Module, Compatibility

6.3 Cambiar: User, Group, DocumentRoot, Port, User, ServerAdmin, ServerName, ServerRoot y ErrorDocument (al menos 403 y 404)

6.4 Ejemplo de Directiva
ServerRoot directive

Syntax: ServerRoot directory-path

Default: ServerRoot /usr/local/apache

Context: server config

Status: core

6.5 Reiniciar Servidor con nueva configuración

7 Configuración Avanzada

7.1 Control de Acceso. Configuración de Directorios (Directiva Directory, Order, Allow, Deny)

7.2 Crear Alias (/imagenes y /manual) (Directiva Alias)

7.3 Archivos log: error.log, access.log (tail -f archivo_log)

7.4 Módulos: Qué son, cómo se añaden y cómo se configuran

7.5 Algunos Módulos: mod_access, mod_auth_digest y mod_mime

8 Problema de la Autenticación: Basada en Host o en Usuario

8.1 Autenticación, Autorización y Control de Acceso

8.2 Directivas: AuthType, AuthName, AuthUserFile, Require

8.3 Autenticación Básica: (htpasswd)

8.4 Autenticación Digest: (htdigest)

7.2 Contenidos Procedimentales

1 Introducción al servidor Web

1.1 Hacer Esquema del proceso de intercambio de una página Web entre Cliente y Servidor

1.2 Buscar alternativas de Servidores Web

2 Instalación

2.1 Configuración, Compilación e Instalación del Servidor (básica)

2.2 Conexión mediante telnet, navegador en modo texto y navegador en modo gráfico

3 Puesta en Marcha

3.1 Revisión de los archivos LOG del servidor

3.2 Interpretación de los datos de los logs

4 Configuración Básica

4.1 Arrancar y Parar el Servidor

4.2 Modificar archivo de configuración (por ejemplo número de puerto, página de error y usuario propietario del servidor)

4.3 Modificar documento raíz del servidor

4.4 Realizar pequeño script que muestre las líneas con errores de archivo no encontrado.

5 Configuración Avanzada

5.1 Configurar permisos a diferentes directorios (por host)

5.2 Configurar alias a los documentos

6 Configuración de la Autenticación

6.1 Acceso con autenticación básica a un directorio

6.1.1 Altas y Bajas de usuarios en el archivo de datos

6.1.2 Configuración de permisos en el directorio

6.2 Acceso con autenticación digest a un directorio

6.2.1 Altas y Bajas de usuarios

6.2.2 Configuración de permisos en el directorio

7.3 Criterios de Evaluación

- Entender el proceso de navegación por Internet dentro de la arquitectura C/S
- Instalar y realizar la configuración básica de un servidor Web
- Realizar el mantenimiento de un servidor Web

Unidad Didáctica 8 Desarrollo de Aplicaciones Web: páginas dinámicas

En el desarrollo de software, las necesidades de la industria pasan cada vez más por el uso de la Plataforma Web. Además, y en coordinación con el módulo de Base de Datos, esta arquitectura permite dar al alumno un concepto integral y completo de una aplicación software. Es además, una de las necesidades que el centro ha detectado de la experiencia de la FCT (Módulo de Formación en Centros de Trabajo).

El uso de aplicaciones web pasa por explicar las diferencias entre páginas estáticas y páginas dinámicas y la relación entre lenguajes interpretados en un Sistema Operativo y el uso del mismo lenguaje interpretado en la arquitectura Web.

8.1 Contenidos Conceptuales

1 Hacer una página HTML interactiva: Formularios

1.1 Qué son (código HTML)

1.2 Para qué sirven (Capa Presentación)

1.3 Cómo funcionan: se rellenan por el usuario y se envían a un programa que los procesa

1.4 Tipos: texto (text y textarea), selección con ratón (checkbox y radiobutton), botones (button), selección de archivos (file), listas de selección(select)

1.5 Sintaxis: Etiquetas (INPUT, atributo NAME y TYPE) dentro de una Etiqueta (FORM, atributo ACTION, METHOD y ENCTYPE)

1.6 Proceso de Envío: ver los campos seleccionados, construir el formulario (incuye todos los campos), se codifica (application/x-www-urlencoded ó multipart/form-data) y se envía (pares ?campo1=valor1&campo2=valor2), enviar el formulario (GET | POST)

1.7 Etiquetas a revisar: FORM, INPUT, SELECT, OPTION y TEXTAREA

2 Introducción a la Programación de Aplicaciones Web

3 Interfaz CGI

3.1 Qué es (Common Gateway Interface)

3.2 Para qué sirve: ejecución de programas en el servidor

3.3 Cómo enviar datos a ese programa (CGI): Relación con los formularios

3.4 Qué necesitamos para utilizar CGIs: programa + configuración servidor + envío de argumentos (formulario)

4 Páginas Dinámicas

4.1 Qué son: programas incluidos en la página Web

4.2 Para qué sirven: ejecución de la lógica de programación en el servidor

4.3 Comparación CGI - Páginas Dinámicas

4.4 Lenguajes: ASP, PHP, Python, SSI, etc

8.2 Contenidos Procedimentales

1. Realización de Formularios

- a) Desarrollar un formulario ejemplo básico
- b) Desarrollar un formulario con todos los tipos de campos que se pueda utilizar como plantilla para el desarrollo de nuevos formularios
- c) Insertar un formulario en una página Web ya hecha

2. Desarrollo de CGIs

- a) CGI básico
- b) Ejecutar un script como un CGI
- c) Configurar adecuadamente los permisos del directorio

3. Página Dinámica (dos clases)

- a) Incluir la programación dentro de la página HTML
- b) Crear un informe, utilizando una página dinámica

8.3 Criterios de Evaluación

→ Ser capaz de realizar una página dinámica sencilla

→ Ser capaz de realizar una página dinámica como respuesta a un formulario

Unidad Didáctica 9 Análisis, Diseño e Implantación de un proyecto Real (Aplicación Web)

Esta unidad didáctica y la siguiente se coordinarán con la asignatura de BD. Tendrá un contenido puramente procedimental desarrollando un proyecto pequeño de desarrollo de una utilidad software a través de Internet. Se denominará mini-proyecto en comparación con el módulo Proyecto Integrado.

Se ofrecerán varias alternativas en forma de requerimientos de un cliente, y los alumnos tendrán que poner en práctica la implementación del sistema de forma autónoma simulando un entorno profesional. Esta unidad didáctica se centrará en la supervisión y repaso de los contenidos conceptuales y procedimentales relacionados con el análisis y diseño de nuevas funcionalidades.

9.1 Contenidos Conceptuales

1 Planificación del Proyecto

1.1 Cómo leer los requisitos

1.2 Análisis de los Requisitos

1.3 Planteamientos de Diseño

1.3.1 Presentación

1.3.2 Datos

1.3.3 Lógica/Programación

2 Diseño y esbozo de la arquitectura a implementar

3 Puesta en marcha de una aplicación Software

3.1 Instalación

3.2 Mantenimiento y control de versiones

3.3 Documentación en línea

9.2 Contenidos Procedimentales

Desarrollo de un mini proyecto a escoger entre cuatro alternativas, o escoger otra alternativa (desde el proyecto integrado o una planteada por el alumno). En grupos de máximo dos personas.

9.3 Criterios de Evaluación

- Ser capaz de montar la arquitectura del sistema
- Ser capaz de realizar el modelo de datos del sistema
- Ser capaz de realizar las páginas dinámicas del sistema

7 CONTENIDOS ACTITUDINALES

Los contenidos relacionados a continuación son comunes a todas las unidades didácticas.

- Comportamiento autónomo para la utilización de bibliografía y otras fuentes de información
- Interés por seleccionar el entorno y las herramientas de programación más adecuados para conseguir mayor productividad
- Interés por utilizar las herramientas de programación más adecuadas para conseguir mayor productividad y portabilidad
- Constancia de la prueba funcional de los programas desarrollados y medida de los recursos del sistema
- Constancia en la documentación del sistema
- Interés por la producción de guías de usuario claras y sencillas
- Valoración de la redacción cuidados de los informes y la utilización adecuada de la terminología informática.
- Hábito de trabajo en grupo y respeto por las normas establecidas en el colectivo.

8 METODOLOGÍA

La metodología que se utilizará en clase potenciará, en lo posible los siguientes aspectos:

- Colaboración y Participación (Internet y clase)
- Interés (Contenidos Interesantes)
- Educación personalizada (Evaluación Continua)

Dentro del proceso de enseñanza-aprendizaje tradicional se pensaba que el conocimiento

fluía del profesor al alumno. Para aumentar la participación e interés del alumno se hace necesario hacerlo intervenir en el proceso. Palabras como “Aprendizaje Significativo” y “Constructivismo” resuenan dentro de las teorías pedagógicas que señalan la participación del alumno en su propio aprendizaje.

El uso de una plataforma de colaboración por Internet permite potenciar la colaboración porque el alumno puede aportar, por ejemplo, una consulta, o un enlace de Internet que ha encontrado o incluso preguntas a otros alumnos. Además el profesor enviará antes de la clase un correo electrónico a todos los alumnos con un pequeño esquema con el mapa de contenidos y las actividades a realizar en clase. De esta forma los servicios por Internet (correo electrónico y navegación Web por el grupo de colaboración) permiten una comunicación fluida entre los alumnos, y con el profesor. Además, y es este uno de los aspectos señalados en el Proyecto Educativo del Centro, el uso de Internet permite una mayor integración de los alumnos que trabajan y tienen dificultades para asistir de forma continua a clase.

Sin embargo este sistema en ningún caso podrá sustituir a la educación tradicional, y en ese sentido, las clases consistirán generalmente en:

- Explicación Oral del mapa de contenidos de la clase, prestando atención a la participación de los alumnos, y aprovechando sus preguntas/intervenciones para aportar los contenidos de la unidad didáctica
- Propuestas de actividades de enseñanza-aprendizaje, de tipo práctico para que el alumno (sólo o en grupo) las desarrolle en clase. Sería ideal también que los alumnos terminen proponiendo alguna actividad.

Además el diseño de la programación y las unidades didácticas presentan contenidos interesantes y actuales, dentro del planteamiento del Diseño Curricular Base del Ciclo Formativo para este módulo. El uso de enlaces y materiales de Internet permite que incluso los alumnos puedan proponer materiales para la unidad didáctica (siempre revisados por el profesor), se generan por lo tanto contenidos más dinámicos que si se utilizara un libro de texto. Hay que tener en cuenta que existirá un trabajo importante de supervisión de los materiales aportados por los alumnos y/o de su programación antes de poner en práctica sus propuestas.

El componente de educación personalizada, es quizá más complejo de delimitar. Para ello, además de la relación con el alumno de tipo personal (en clase y vía correo electrónico) se tratará de, en cada unidad didáctica, evaluar de forma personalizada e incluso asignar tareas de formas personalizada. Para ello se utilizarán los criterios de clasificación de muy bien, bien, regular y mal en la evaluación práctica continua que permitirán conocer el grado de soltura en el manejo de contenidos de la unidad didáctica, que tiene el alumno. Estos aspectos se dejarán claros al comienzo del módulo tratando de aclarar que a cada alumno se le evaluará de forma personalizada según su nivel pero con pruebas generales (test) donde la evaluación será relativa a la clase.

Esta metodología pretende tener al alumno con una dinámica de estudio y trabajo continuo, sobre todo en clase.

9 MEDIDAS DE ATENCIÓN A LA DIVERSIDAD.

Las medidas de refuerzo educativo tendrán como objetivo potenciar los conocimientos y procedimientos englobados en los contenidos mínimos exigibles, que no fueron superados por los alumnos en las diferentes pruebas de evaluación. Para estos alumnos se realizarán las siguientes medidas:

- Ejercicios específicos
- Atención exclusiva con herramientas síncronas (Chat y Telefonía IP) y asíncronas (correo electrónico y Grupo Yahoo)
- Evaluación diaria con mayor supervisión

El profesor mantendrá un servicio Web de la asignatura (<http://www.cesareox.com/docencia/dfsif/>), con la información del curso, enviará un correo electrónico con el esquema de la clase y permitirá a estos alumnos comunicarse vía correo electrónico o chat según disponibilidad. Además el centro mantiene para cada ciclo un grupo yahoo de colaboración.

También existirán alumnos que obtendrán rápidamente los conocimientos y procedimientos de la unidad didáctica, ya sea por su capacidad o por los conocimientos autodidactas tan comunes en las áreas de informática. Para estos alumnos se tendrán en cuenta las siguientes medidas:

- Profundizarán en cada unidad didáctica y se hará supervisión
- Generarán documentación Web para el resto de alumnos

En el ámbito de la Formación Profesional, cada vez más coordinada con la realidad socio laboral, existen alumnos (especialmente en los turnos de tarde y noche) que trabajan y se les hace complicado asistir a todas las sesiones. Esta situación no es problema gracias al uso de herramientas Internet.

Además el uso de Internet y una plataforma de colaboración es de hecho la medida más importante de atención a la diversidad, pues permite que se potencia la relación entre alumnos que o no pueden asistir a clase o tienen dificultades en el seguimiento.

10 Evaluación

10.1 Procedimiento de Evaluación Ordinario

El sistema de evaluación empleado tiene como objetivo evaluar los contenidos, procedimientos y actitudes que pretendemos que el alumno tenga, pero además queremos transmitir la idea de la disciplina y la autoformación.

Por otra parte, basaremos la evaluación como una forma de supervisión en el mundo laboral y se permitirá en los exámenes el acceso a apuntes, notas, Internet o cualquier medio de documentación disponible también en el mundo laboral.

La evaluación continua es la esencia de este sistema de evaluación, ayudando a tener un ritmo de trabajo todo el curso.

Las medidas de evaluación serán, para cada evaluación las mismas. Recordemos que este módulo tiene dos evaluaciones. El sistema se basará en:

- Al menos tres exámenes tipo test de 40 preguntas, con cuatro alternativas y una respuesta única en cada pregunta. Los exámenes serán sorpresa e incluirán cualquier unidad didáctica (o parte de ella) dada en clase hasta el día anterior. La nota de cada alumno se comparará con el rendimiento de la clase en el examen. Será el 30% de la nota. La nota será entre 1 y 10, con un decimal.
- Se hará una evaluación diaria del trabajo práctico y procedimental de forma continua obteniendo entre 6 y 8 notas a lo largo de una evaluación. El profesor podrá evaluar en cualquier momento. Será el 30% de la nota y será uno de los siguientes valores: Muy Bien, Bien, Regular y Mal. Se podrán pedir trabajos prácticos o tareas específicas al alumno, o grupo de alumnos fruto de este tipo de evaluación.
- Al final de la evaluación existirá un examen oral y personal para cada alumno. Tendrá una duración de 15 minutos y se podrá preguntar a cada alumno cualquier contenido teórico y práctico del módulo. Tendrá un valor del 30% de la nota y será una nota entre 1 y 10, con un sólo decimal. En la última evaluación este examen será un proyecto final.
- Valoración del trabajo en clase, la asistencia y la colaboración con otros alumnos. Esta nota supone el 10% del total y el profesor evaluará la participación en clase, la colaboración en la plataforma de Internet y la asistencia a las sesiones. Será una nota entre 1 y 10 con un decimal

No existirá el concepto de “recuperación” en el sentido de que existan exámenes específicos cuando no se superen unidades anteriores. Se hará una evaluación continua y personalizada, que permite evaluar al alumno todos los contenidos. No se dividirá en

partes el módulo y no existirá el concepto de “unidades pendientes” porque se pretende transmitir al alumno un concepto integral del módulo.

En las evaluaciones se utilizarán dos tipos de notas, entre 1 y 10 con un decimal y uno entre cuatro valores (Mal, Regular, Bien y Muy Bien). En el área de la administración de sistemas informáticos la industria suele categorizar a los profesionales en Novato, Junior, Senior y Experto, por ellos hemos utilizado las notas como una forma de evaluar al alumno suponiendo una contratación del alumno por parte de la empresa “profesor”. De esta manera la correspondencia de notas sería:

- 10 y 9. Estos alumnos pueden trabajar sin casi supervisión, son capaces incluso de tomar decisiones sobre nuevas tareas y manejar el concepto evaluado con soltura.
- 8. Estos alumnos tienen bastante soltura pero precisan de supervisión, no mucha pero necesitarían formación extra de corta duración para adquirir una soltura total.
- 7 Estos alumnos conocen las tareas y las pueden desarrollar pero necesitan de revisión muy detallada
- 5 y 6 Estos alumnos tienen los conocimientos mínimos pero deberían tener más práctica en las tareas y competencias del módulo. Para adquirir soltura necesitarían formación de larga duración.

En el caso de pérdida de evaluación continua, por ejemplo si las faltas de asistencia son mayores al 10% de las sesiones del módulo, se hará un examen final que constará de:

- Examen tipo test de 120 preguntas. Será eliminatorio y el alumno para aprobar debe sacar un mínimo de 70 respuestas correctas. Tendrá una duración de 100 minutos.

- Examen práctico de 4 horas de duración para evaluar contenidos procedimentales, delante del ordenador
- Examen oral y personal de 30 minutos para la evaluación de todo el módulo

10.2 Criterios y Procedimientos para establecer las actividades de recuperación de los módulos pendientes

La recuperación de este módulo, cuando esté pendiente se realizará en el último trimestre del segundo curso. El profesor encargado de las tareas de recuperación y evaluación será un profesor de segundo curso.

El alumno que lo desee como medida de refuerzo podrá asistir a clase siempre y cuando no interrumpa ni altere las clases del módulo de segundo curso y exista espacio físico en el aula. Si el alumno se incorpora al módulo, al repetir curso, se utilizarán los criterios de evaluación ordinarios

El sistema de recuperación se hará basándose en el trabajo personal del alumno y la consulta en horas de profesor en el último trimestre del curso. Las horas disponibles para recuperación serán aproximadamente la mitad de las del curso ordinario.

El sistema de recuperación se basará en el desarrollo práctico de un miniproyecto, equivalente a las dos últimas unidades didácticas y la evaluación continua de contenidos de las unidades didácticas a medida que avanza el trimestre.

10.3 Procedimientos de Evaluación y cualificación extraordinaria del alumnado con módulos pendientes

Para aprobar el módulo deberán ser capaces de realizar la configuración básica y la instalación del Servidor Web con una página Web. La nota vendrá dada por el desarrollo del mini proyecto que se presenta en las dos últimas unidades didácticas del módulo.

Se evaluará este trabajo y se harán evaluaciones periódicas de tipo práctico para evaluar los contenidos procedimentales y conceptuales de las unidades didácticas.

10.4 Contenidos Mínimos Exigibles

El procedimiento de evaluación tratará de medir el alcance de los objetivos del módulo. Para cada unidad didáctica se seguirá el procedimiento de evaluación descrito en los apartados anteriores, teniendo en cuenta además el criterio de notas que nos hemos marcado.

Deste esta postura, un alumno, para aprobar, es decir para obtener un 5, ha de tener los conocimientos mínimos de las capacidades terminales del módulo. Para ello se evaluarán los contenidos de cada unidad didáctica (procedimentales, conceptuales y actitudinales) como al resto de los alumnos.

El contenido mínimo significa que el alumno tiene un conocimiento básico del contenido a evaluar, aunque no tenga la soltura suficiente para tomar decisiones sobre nuevas tareas. El criterio de las notas de evaluación sobre los contenidos se ha explicado en el apartado sobre criterios de evaluación.

11 Bibliografía

La bibliografía se apoyará fundamentalmente en recursos Web que se irán actualizando a lo largo del módulo por los alumnos y el profesor. La creación de recursos dinámicos en un área que evoluciona tan rápido como la informática es relativamente sencillo con el uso de Internet.

El uso de Internet como fuente de recursos para el módulo, se incluye dentro del planteamiento general del centro descrito en el PEC y PCC. Esta situación va evolucionando a medida que avanza el curso, aunque algunas de las propuestas iniciales son revisar en sites¹ como los que mostramos a continuación:

- <http://www.tldp.org> Linux Documentation Project. Documentación de Manuales, formas de documentación, procesos y sistemas operativos (es.tldp.org es el proyecto de traducción)
- <http://www.w3c.org> World Wide Web Consortium. Estándares y Arquitectura Internet
- <http://www.apache.org> Arquitectura Internet, configuración e instalación de Servidor Web
- <http://www.microsoft.com/> Arquitectura Windows y DOS, Ayuda de Sistemas operativos y librerías

¹ Ponemos el nombre del site no las páginas en concreto.

Se utilizarán también los manuales de usuario, de administrador y las ayudas instaladas localmente para cada Sistema Operativo y las aplicaciones software que utilicemos. Especialmente para cada lenguaje de programación se utilizará la ayuda que ofrezca la comunidad de soporte.

También se utilizarán las especificaciones del lenguaje utilizando la documentación pública, por ejemplo RFCs y especificaciones de W3C e IETF. Sin embargo no se recomendará la compra de ningún libro.

Algunos libros interesantes de consulta y que permiten profundizar más en los contenidos son:

Herbert Schildt. Manual de referencia. McGrawHill.

S. Prata/ D. Martin. Unix Sistema V. Anaya Multimedia.

Kernighan Pike. El entorno de programación Unix. Prentice Hall

Jakob Nielsen, Designing Web Usability. ISBN 1-56205-810-X. Diciembre 1999 Ed New Riders

Philip Greenspun. Philip and Alex's Guide to Web Publishing ISBN 1-55860-534-7. Año 1999. Ed Morgan Kaufmann